80/411/FDIS

FINAL DRAFT INTERNATIONAL STANDARD
PROJET FINAL DE NORME INTERNATIONALE

	Project number Numéro de projet	**IEC 61162-402 Ed.1**
	IEC/TC or SC CEI/CE ou SC **TC 80**	Secretariat / Secrétariat **United Kingdom**
☒ Submitted for parallel voting in CENELEC Soumis au vote parallèle au CENELEC	Distributed on / Diffusé le **2005-06-17**	Voting terminates on / Vote clos le **2005-09-02**

Also of interest to the following committees Intéresse également les comités suivants	Supersedes document Remplace le document 80/400/CDV - 80/409/RVC

Functions concerned
Fonctions concernées

□ Safety Sécurité	□ EMC CEM	□ Environment Environnement	□ Quality assurance Assurance de la qualité

INTERNATIONAL ELECTROTECHNICAL COMMISSION **COMMISSION ÉLECTROTECHNIQUE INTERNATIONALE**

THIS DOCUMENT IS A DRAFT DISTRIBUTED FOR APPROVAL. IT MAY NOT BE REFERRED TO AS AN INTERNATIONAL STANDARD UNTIL PUBLISHED AS SUCH.

IN ADDITION TO THEIR EVALUATION AS BEING ACCEPTABLE FOR INDUSTRIAL, TECHNOLOGICAL, COMMERCIAL AND USER PURPOSES, FINAL DRAFT INTERNATIONAL STANDARDS MAY ON OCCASION HAVE TO BE CONSIDERED IN THE LIGHT OF THEIR POTENTIAL TO BECOME STANDARDS TO WHICH REFERENCE MAY BE MADE IN NATIONAL REGULATIONS.

CE DOCUMENT EST UN PROJET DIFFUSÉ POUR APPROBATION. IL NE PEUT ÊTRE CITÉ COMME NORME INTERNATIONALE AVANT SA PUBLICATION EN TANT QUE TELLE.

OUTRE LE FAIT D'ÊTRE EXAMINÉS POUR ÉTABLIR S'ILS SONT ACCEPTABLES À DES FINS INDUSTRIELLES, TECHNOLOGIQUES ET COMMERCIALES, AINSI QUE DU POINT DE VUE DES UTILISATEURS, LES PROJETS FINAUX DE NORMES INTERNATIONALES DOIVENT PARFOIS ÊTRE EXAMINÉS EN VUE DE LEUR POSSIBILITÉ DE DEVENIR DES NORMES POUVANT SERVIR DE RÉFÉRENCE DANS LES RÈGLEMENTATIONS NATIONALES.

Title

IEC 61162-402 Ed.1:Maritime navigation and radiocommunication equipment and systems - Digital interfaces - Part 402: Multiple talkers and multiple listeners - Ship systems interconnection - Documentation and test requirements

Titre

ATTENTION
VOTE PARALLÈLE
CEI – CENELEC

L'attention des Comités nationaux de la CEI, membres du CENELEC, est attirée sur le fait que ce projet final de Norme internationale est soumis au vote parallèle. Un bulletin de vote séparé pour le vote CENELEC leur sera envoyé par le Secrétariat Central du CENELEC.

ATTENTION
IEC – CENELEC
PARALLEL VOTING

The attention of IEC National Committees, members of CENELEC, is drawn to the fact that this final Draft International Standard (DIS) is submitted for parallel voting. A separate form for CENELEC voting will be sent to them by the CENELEC Central Secretariat.

CONTENTS

INTERNATIONAL ELECTROTECHNICAL COMMISSION

MARITIME NAVIGATION AND RADIOCOMMUNICATION EQUIPMENT AND SYSTEMS – DIGITAL INTERFACES –

Part 402: Multiple talkers and multiple listeners – Ship systems interconnection – Documentation and test requirements

FOREWORD

1) The International Electrotechnical Commission (IEC) is a worldwide organization for standardization comprising all national electrotechnical committees (IEC National Committees). The object of IEC is to promote international co-operation on all questions concerning standardization in the electrical and electronic fields. To this end and in addition to other activities, IEC publishes International Standards, Technical Specifications, Technical Reports, Publicly Available Specifications (PAS) and Guides (hereafter referred to as “IEC Publication(s)”). Their preparation is entrusted to technical committees; any IEC National Committee interested in the subject dealt with may participate in this preparatory work. International, governmental and non-governmental organizations liaising with the IEC also participate in this preparation. IEC collaborates closely with the International Organization for Standardization (ISO) in accordance with conditions determined by agreement between the two organizations.
2) The formal decisions or agreements of IEC on technical matters express, as nearly as possible, an international consensus of opinion on the relevant subjects since each technical committee has representation from all interested IEC National Committees.
3) IEC Publications have the form of recommendations for international use and are accepted by IEC National Committees in that sense. While all reasonable efforts are made to ensure that the technical content of IEC Publications is accurate, IEC cannot be held responsible for the way in which they are used or for any misinterpretation by any end user.
4) In order to promote international uniformity, IEC National Committees undertake to apply IEC Publications transparently to the maximum extent possible in their national and regional publications. Any divergence between any IEC Publication and the corresponding national or regional publication shall be clearly indicated in the latter.
5) IEC provides no marking procedure to indicate its approval and cannot be rendered responsible for any equipment declared to be in conformity with an IEC Publication.
6) All users should ensure that they have the latest edition of this publication.
7) No liability shall attach to IEC or its directors, employees, servants or agents including individual experts and members of its technical committees and IEC National Committees for any personal injury, property damage or other damage of any nature whatsoever, whether direct or indirect, or for costs (including legal fees) and expenses arising out of the publication, use of, or reliance upon, this IEC Publication or any other IEC Publications.
8) Attention is drawn to the Normative references cited in this publication. Use of the referenced publications is indispensable for the correct application of this publication.
9) Attention is drawn to the possibility that some of the elements of this IEC Publication may be the subject of patent rights. IEC shall not be held responsible for identifying any or all such patent rights.

IEC 61162-402 has been prepared by Technical Committee 80: Maritime navigation and radiocommunication equipment and systems.

The text of this CDV is based on the following documents:

FDIS	Report on voting
80/XX/FDIS	80/XX/RVD

Full information on the voting for the approval of this standard can be found in the report on voting indicated in the above table.

This publication has been drafted in accordance with the ISO/IEC Directives, Part 2.

IEC 61162 consists of the following parts, under the general title *Maritime navigation and radiocommunication equipment and systems – Digital interfaces*:

Part 1: Single talker and multiple listeners

Part 2: Single talker and multiple listeners, high-speed transmission

Part 400: Multiple talkers and multiple listeners – Ship systems interconnection – Introduction and general principles

Part 401: Multiple talkers and multiple listeners – Ship systems interconnection – Application profile

Part 402: Multiple talkers and multiple listeners – Ship systems interconnection – Documentation and test requirements

Part 410: Multiple talkers and multiple listeners – Ship systems interconnection – Transport profile requirements and basic transport profile

Part 420: Multiple talkers and multiple listeners – Ship systems interconnection – Companion standard requirements and basic companion standards

The committee has decided that the contents of this publication will remain unchanged until the maintenance result date[1] indicated on the IEC web site under "http://webstore.iec.ch" in the data related to the specific publication. At this date, the publication will be

- reconfirmed;
- withdrawn;
- replaced by a revised edition, or
- amended.

A bilingual version of this publication may be issued at a later date.

[1] The National Committees are requested to note that for this publication the maintenance result date is 2008.

INTRODUCTION

IEC 61162 is a four part standard which specifies four digital interfaces for applications in marine navigation, radio-communication and system integration.

The four parts are:

IEC 61162-1	Single talker and multiple listeners
IEC 61162-2	Single talker and multiple listeners – High speed transmission
IEC 61162-3	Multiple talkers and multiple listeners – Serial data instrument network
IEC 61162-4	Multiple talkers and multiple listeners – Ship systems interconnection. This part is sub-divided into a number of individual standards with part numbers in the IEC 61162-400 series.

This part of the standard, IEC 61162-402, Documentation and test requirements, defines the minimum documentation and test requirements for implementations of the standard.

MARITIME NAVIGATION AND RADIOCOMMUNICATION EQUIPMENT AND SYSTEMS – DIGITAL INTERFACES –

Part 402: Multiple talkers and multiple listeners – Ship systems interconnection – Documentation and test requirements

1 Scope

1.1 General

This standard series, IEC 61162-400 and upwards, specifies a communication protocol for use in integrated ship systems. It also specifies an interface description language for use together with the protocol, a set of rules for the use of this language and a set of standard interfaces described in the language.

This part of the standard specifies a minimum set of tests to be done, test results to be achieved and documents that shall be available for all implementations of general protocol software and applications that are compliant with the IEC 61162-4 standard. Although this set of standard documents is collectively referred to as IEC 61162-4, the actual part numbers are in the 400-series (see 1.4 of IEC 61162-400).

1.2 Limitations in scope

The tests and documentation requirements do not cover electrical, physical or environmental requirements that may apply to the use of software or computers onboard ships. Such requirements may be covered by IEC 60945 or IEC 60092-504. Other standards may also be applicable.

This standard does not necessarily cover all requirements from classification societies or other authorities. It is the responsibility of the user of this standard to ensure that all appropriate regulations are addressed.

This standard contains tests to check that an application using the IEC 61162-4 protocol adheres to its advertised interface specification. These tests cannot guarantee the correct functionality of that application beyond the possibility of connecting it to the network and with a limited degree of accuracy in the messages transferred.

This standard does not cover the system in which the IEC 61162-4 communication standard is used. Additional requirements will normally apply to the total system configuration.

Fundamental requirements relating to ensuring reliable and timely transfer of data across data communication links are included in other standards associated with the integration of equipment such as IEC 60092-504 and IEC 61209. This standard does not contain tests to verify compliance with these requirements. In addition, specific equipment related standards may also contain requirements for correctness and timeliness of data transmissions. Neither does this standard contain any tests to verify such requirements. Thus, results from tests carried out in accordance with this standard cannot be used to demonstrate compliance with the requirements of any other standards for system or equipment functionality.

1.3 Limitations in test coverage

The test plan only specifies general tests of the protocols and a limited set of other general properties (black box tests). The test procedures will not generally cover tests of operating systems, communication libraries or other software components that are used to implement the standard. Neither does this standard specify any tests related to the way the system is implemented (white or glass box testing).

1.4 Limitations in degree of detail

The test procedures are general in nature and do not generally specify detailed test programs and procedures. The procedures specify a minimum set of functional aspects that need to be tested, with, in some cases, a minimum required set of excitations and corresponding required responses. The testers must develop the detailed procedures and test tools themselves.

2 Normative references

The following referenced documents are indispensable for the application of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

IEC 60092-504, *Electrical Installations in ships – Special features – Control and instrumentation*

IEC 60945, *Maritime navigation and radiocommunication equipment and systems – General requirements – Methods of testing and required test results*

IEC 61162-4, *(shorthand for all parts in the IEC 61162-400 series), Maritime navigation and radio-communication equipment and systems – Digital interfaces – Part 4xx: Multiple talkers and multiple listeners – Ship systems interconnection*

IEC 61162-400, *Maritime navigation and radiocommunication equipment and systems – Digital interfaces – Part 400: Multiple talkers and multiple listeners – Ship systems interconnection – Introduction and general principles*

IEC 61162-401, *Multiple talkers and multiple listeners – Ship systems interconnection – Application profile*

IEC 61162-410, *Multiple talkers and multiple listeners – Ship systems interconnection – Transport profile requirements and basic transport profile*

IEC 61162-420, *Multiple talkers and multiple listeners – Ship systems interconnection – Companion standard requirements and basic companion standards*

IEC 61209, *Maritime navigation and radiocommunication equipment and systems – Integrated Bridge Systems (IBS) – Operational and performance requirements, methods of testing and required test results*

IEC 61508-3, *Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 3: Software requirements.*

IEC 61508-4*, Functional safety of electrical/electronic/programmable electronic safety-related systems – Part 4: Definitions and abbreviations*

ISO 9001: 2000, *Quality management systems – Requirements.*

ISO/IEC 90003: 2004, *Software engineering – Guidelines for the application of ISO 9001: 2000 to computer software.*

3 Definitions

For the purposes of this document, the following definitions apply.

3.1
black-box testing
testing that ignores the internal workings and internal structure of a component and focuses on the responses generated as a result of controlled stimuli and execution conditions. Typically used to evaluate the compliance of a component with specified functional requirements. See also *white-box testing*

3.2
defect
latent faults in a component ("bug" in software), that either represent or can cause an error and by that a failure

3.3
error
that part of the system state that is liable to lead to a failure (IEC 61508-4) IEC 61508-4 does not classify a software defect as an error, but as a fault. In this standard, the term defect will be used to mean also software defects. The term fault will not be used.

3.4
fault
see error and defect

3.5
failure
occurs when a delivered service deviates from the intended service. It is the effect of an *error* on the service (IEC 61508-4)

3.6
memory leak
situation where a program is not able to reclaim dynamically allocated memory that should be released as a result of the removal of an internal object. It typically occurs during sequences of connect and disconnect

3.7
safety integrity level
discrete level (one out of a possible four) where safety integrity level 4 has the highest level of safety integrity and safety integrity level 1 has the lowest. Safety integrity is the probability of a safety-related system satisfactorily performing the required safety functions under all the stated conditions within a stated period of time (see IEC 61508-4).

3.8
white-box testing
testing that uses knowledge of the internal structure and internal workings of a component to exercise, for example selected internal execution paths or sub-component interactions in the component. See also *black-box testing*.

4 Overview and basic principles

4.1 Introduction

This part of IEC 61162 covers test and documentation requirements. Proper testing, based on a test plan, and the availability of documentation are factors that are important in ensuring the correctness of a protocol or application software module. This document specifies general requirements to testing and documentation for both protocol and application modules. This document only specifies the tests that have to be made and the required test results. It does not specify the tools or mechanisms that are used to perform the test. This is the responsibility of the tester.

Documentation requirements are more specific and define the minimum requirements for documentation that follows protocol or application modules. The user should take care to supplement the minimum requirements with whatever extra documentation that it is felt to be necessary to use the module in question. Of particular importance is software documentation in the case where there is the possibility to modify the module.

Annexes summarise the test requirements in a form that can be used as a test log.

4.2 Purpose of this standard

This standard shall help to ensure that important aspects of an implementation of IEC 61162-4 basic software does what it is supposed to do and that it does not contain any hidden defects. This standard can also be used to ensure that an application using the IEC 61162-4 standard actually implements the interface to the network that it advertises through its specification or companion standard document.

This standard shall also define a minimum set of documents that shall follow the application or be available from the developer of the application or communication software. These documents will partly specify interface and functionality attributes as well as act as part proof of the implementation's adherence to the IEC 61162-4 specification.

With these two goals in mind, this standard covers part of the verification and validation process that is necessary to produce safe integrated ship systems. The main emphasis is, however, on verification.

4.3 Use in the different stages of a development process

The stages of a development process are dependent on the process being used and how that process is implemented. However, the stages on a high level can be characterised as belonging to the specification, design, implementation and integration phases. The following clauses will, [with the basis] in these phases, specify where this standard can be applied and which other standards can be used.

This standard does not address the software development- and lifecycle as such. However, this standard requires that any software produced to comply with IEC 61162-4, as a minimum is developed to the ISO 9001 standard and implements the relevant part of this standard as specified in ISO/IEC 90003, for the software product, or to equivalent standards.

4.3.1 Specification

The specification of an IEC 61162-4 module is contained in IEC 61162-400, IEC 61162-401 and IEC 61162-410. The interface between applications and the IEC 61162-4 network shall be specified through companion standard documents as prescribed in IEC 61162-420.

4.3.2 Design

IEC 61162-400, IEC 61162-401 and IEC 61162-410 contain parts of the design specification in the form of ER-diagrams, message sequence charts, state diagrams and basic modularisation. Additional design documents are, however, necessary for the coding of an IEC 61162-4 implementation. This standard does not prescribe particular methods or tests for the preparation of design documents.

The IEC 61508-3 standard may be appropriate for certain types of system that need a high safety integrity level. The standard will, in any case, contain guidelines that can be used in the design phase.

4.3.3 Implementation

No part of this standard prescribes any particular principle that shall be used during implementation of IEC 61162-4 compliant devices.

IEC 61508-3 may be appropriate for certain types of system that need a high safety integrity level. The standard will in any case contain guidelines that can be used in the implementation phase.

4.3.4 Integration

This part of IEC 61162 describes a set of functional tests that shall be performed on a finished IEC 61162-4 module or application. Some of these tests are appropriate as pre-integration tests and can also be helpful in pinpointing particular problems in the implementation. Notes in the standard will give information to that effect, where appropriate.

IEC 61508-3 may be appropriate for certain types of systems that need a high safety integrity level. This standard will in any case contain guidelines that can be used in the integration phase.

IEC 61209 also contains requirements that are appropriate for certain types of systems, in particular integrated bridge systems.

4.3.5 Verification

This standard covers functional tests (black-box tests) that shall be used to verify that a module, or an implementation thereof, using the IEC 61162-4 protocol, satisfies certain functional requirements that are inherent in the test section of this part of the standard. This standard is mainly intended for the use in the verification phase.

4.4 Structure of this standard

This clause specifies general requirements of the development process. Clause 5 identifies the critical functionality in the IEC 61162-4 protocol and relevant test scenarios. Clause 6 defines test tools and test scenarios. Clause 7 contains test plans for general protocol modules. Clause 8 contains test plans for application modules. Clause 9 contains documentation requirements. Annexes contain summary tables that can be used as basis for the creation of test and documentation logs and check lists.

5 Critical functionality in the protocol

This clause analyses the typical IEC 61162-4 functionality and system architecture and defines the most important test scenarios. The purpose of this clause is to describe the rationale behind the selection of test cases and also to be a basis for the creation of more extensive and voluntary tests when these are desired by the implementers or users.

5.1 Function groups

An implementation of the IEC 61162-4 protocol standard will typically have to handle a set of different functions where an error in any or each can cause failure. The most important functions are listed in the following clauses.

5.1.1 MAU management

A MAU and an LNA must be able to co-operate to establish a MAU-LNA session and provide, for example MAU name services and MAU watchdog functions. The typical stages in MAU management are:

a) Accept a connection attempt from a MAU and register that MAU as existing in the system. This includes checks for duplicate MAU names and the two connection sequences that need to be considered: 1) LNA starts before MAU. 2) MAU starts before LNA.

b) Make the new MAU name and status available in the network. Respond to messages about other MAUs by providing additional information, for example about duplicate names.

c) Provide the optional watchdog function and, if necessary, let the LNA kill the MAU when the watchdog fails.

d) From the LNA, handle the death of a MAU correctly, i.e. clean up internal state and report death to the system.

e) From the MAU, handle the death of the LNA correctly, i.e. clean up internal state and start reconnection attempts if appropriate. This also applies to the closing down of the MAU-LNA link from the LNA.

f) Handle the reconnect of a previously dead MAU or LNA correctly.

5.1.2 Interface and session management

MAUs connect to each other through interfaces. The system must be able to handle the establishment and disruption of such connections as prescribed in the A-profile. The system must also be able to handle session management, i.e. identification of parties in an exchange of messages and flow control. The cases that need to be considered are enumerated below:

a) A server MAU exports an interface for use by clients. Note that there is a time difference between the establishment of the server MAU session and the export of the interface and that this must be handled by the LNA when remote clients attempt to connect.

b) A client MAU connects to the interface.

   Note that steps a) and b) may be executed in the opposite order.

c) The server LNA shall check the connect message and, if appropriate, send a connection request to the server MAU. Checks shall be made that the client is allowed to connect and that the client's request is a true sub-set of the servers advertised interface.

d) If the connection attempt is accepted by both server components, i.e. the LNA and MAU, an acknowledgement shall be sent to the client MAU. The client can start to send transaction requests.

e) The server LNA and MAU shall be able to handle multiple clients in the same manner and be able to keep the different sessions apart with regard to transaction source identity and routing of transactions.

f) The client MAU or its LNA may die or the client MAU may close its side of the connection. The server MAU shall be notified of the closing and the LNA clean up internal state, including discarding any pending transactions.

g) The server MAU or its LNA may die or the server MAU may close its side of the connection. The client shall be notified and the LNA clean up its internal state.

   Note that f) may occur before g) or the two may occur at the same time.

h) The client and the server shall be able to reconnect at any time and the connection shall be re-established as for the first connection. The client shall, if appropriate, be automatically reconnected by the LNA.

Connection management must handle an arbitrary large number of clients and server MAUs in all possible configurations; also when both client and server are located at the same LNA.

5.1.3 Transaction management

Data is exchanged as transactions between a client and server MAU. The system must be able to execute these transactions correctly and on time. Transactions are performed in a number of distinct steps:

a) The client MAU creates a request.

b) The client MAU protocol library (MAPI) adds address information and converts the outgoing message to network format (data marshalling).

c) The client LNA multiplexes outgoing messages to correct destination LNA.

d) The server LNA de-multiplexes incoming messages to correct MAU.

e) The server MAU's MAPI converts the message to internal format, marshals the data and extracts address information.

f) An application routine in the server MAU processes the message and generates an acknowledgement. Multiple acknowledgements may also be generated for subscription messages.

g) Address information is added to the acknowledgement by the MAPI and the message is converted to network format.

h) The server LNA identifies the target MAU and multiplexes the message onto the correct LNA link. For some message types (subscription), the LNA must duplicate the message to a number of subscribers.

i) The client LNA receives the message and targets it at the client MAU.

j) The client MAPI converts the message to internal format and passes it to the correct application handler routine.

k) The application part of the MAU processes the message.

In addition to this, all parties involved must be able to handle a transaction cancellation issued at any time in the sequence. The system must also handle the shutting down of a connection, by a command or as the result of a connection failure, at any time in the sequence.

Transaction management must handle an arbitrary large number of client and server MAUs in all possible configurations, also when both client and server are located at the same LNA.

5.2 High loading and general exception handling

The system shall also be able to handle abnormal situations that occur due to high load or physical problems in the system. It is also necessary to quantify any load related effects that may occur in the system.

5.2.1 Session limitation

The MAU shall be able to send session control messages to another MAU in the system. These messages shall inhibit transmission of non-urgent data.

A server MAU can limit the number of clients that can connect to an interface. The server LNA enforces this limit.

5.2.2 Load limitation

A server MAU can specify a maximum number of pending transactions on an interface. Excessive non-urgent transactions are denied by the LNA.

Urgent transactions shall in any case have priority before non-urgent transactions.

5.2.3 Load tests

The IEC 61162-4 protocol, with Ethernet based T-profile, will normally be limited in its network performance by the CPU power. It is in the particular context of switches between LNA and MAUs that may cause high loads and, thus, delays in the system. It is necessary to quantify this performance degradation.

In some cases, a system may also be limited by network or computer input/output bandwidth. It is also necessary to quantify this effect where it occurs.

5.2.4 Exception handling

The system shall tolerate physical errors in the system. The cases that need consideration are:

a) Sudden death of a host computer, including sudden loss of a communication link. This makes it impossible to shut down the communication link properly and the system may be dependent on link watchdogs to detect the failure. This is a particular problem when the link is idle most of the time.

b) Errors on the hardware link interface that may give the host computer problems, for example loss of carrier, may cause system software lock-up. Excessive interrupts can cause high load on the computer.

c) Loss of redundancy. The system shall continue operation without any loss of functionality. Warnings about loss of redundancy must be given to higher level error handlers. Various transitions between redundant and non-redundant must be handled.

5.3 Generalised architecture

An IEC 61162-4 system can consist of any number of MAUs that in turn are assigned to a number of LNAs. Each LNA must run on a separate host computer, but the MAUs may be distributed between LNAs as is most convenient. A generalisation of the typical communication paths in such a system is illustrated below.

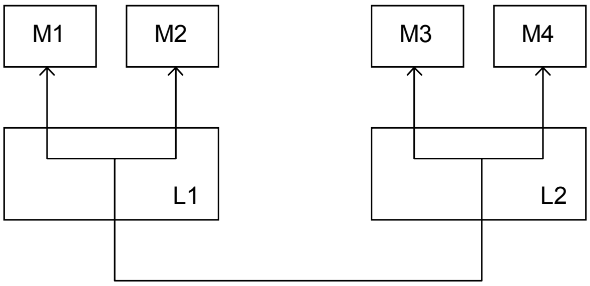

Figure 1 – Typical communication paths in an IEC 61162-4 system

Each LNA (L1 and L2) must multiplex data from and to each of its MAUs (M1 to M4). A real system will typically consist of many more LNAs and usually more MAUs per LNA. However, the possible faults that can occur can be generalised in this 4 times 2 diagram. The cases that will have to be checked are:

a) M1 is a client of itself (special case, but legal).

b) M1 is a client of M2 (same LNA).

c) M1 is a client of M3 (different LNA).

d) M1 and M2 are clients of M3 and M4 (multiplex needed on both client and server side).

e) M1, M2 and M3 are clients of M4 (L2 need to send some messages remote and some locally).

f) M1 is a client of M2, M3 and M4 (same as previous, but test on the receiving side).

The above cases are the ones that need careful testing with respect to relevant functionality and exception handling. These cases shall also be checked with larger number of LNAs and MAUs , but in these cases one need only verify that the basic functionality is present.

5.4 Message passing contribution to possible errors

A typical message transfer can be illustrated as in the following figure:

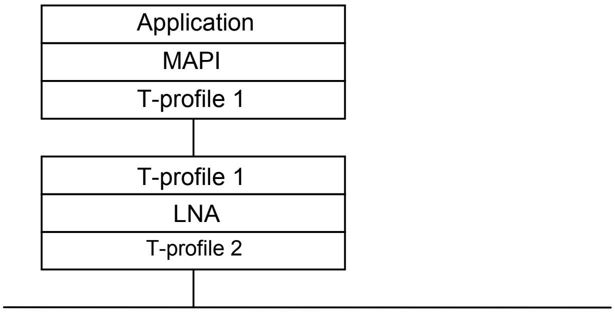

Figure 2 – Message flow in IEC 61162-4 system

The MAU consists of the application part, the MAPI library and a T-profile library for passing data to the LNA. The LNA consists of a T-profile library for communication with the MAU, the actual message processing code and a T-profile for communication with other LNAs. Messages must generally pass down through these layers on the way down from the MAU and up through the same layers on the remote side of the link.

Basically, there is only a limited set of failures that can occur at the message passing level:

a) The message may initially get bad contents due to errors during the message preparation phase.

b) The message may be garbled due to errors in data marshalling (conversion to and from network data format).

c) The message may be routed the wrong way due to errors in message management functions.

d) The message contents may be destroyed due to errors in software or hardware (overwritten, truncated, etc.).

Based on the functional analysis one can define places in the architecture that have a high likelihood of containing errors. The above list covers MAU management, connection establishment and transaction handling, as all functions to a large degree, consist of the same primitive operations, i.e. Message passing with consistency in addressing and content.

6 Test tools and test scenarios

6.1 Reference topology

The following figure defines the topology and the modules that are referenced in this standard:

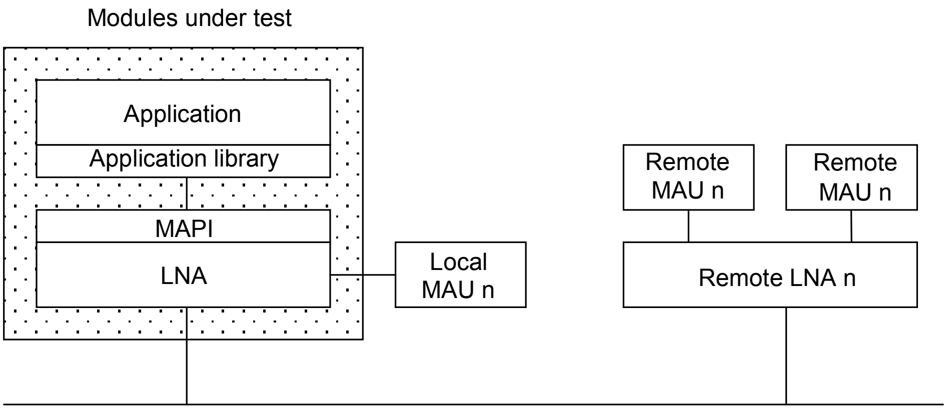

Figure 3 – Test topology

The modules that are tested are within the shaded rectangle. The modules are:

a) Application: application software using the services of the protocol module (MAPI and LNA) to perform some function in a distributed control and monitoring system. This is an application module.

b) Application library: component library, possibly produced by a third party, that implements some more or less general functions for the application. Typical examples are the general MAU management functions (`PACFullApplication` interface components from IEC 61162-420) or a general data transmission facility (e.g. `PACServerApp`). An application library shall adhere to the same requirements as an application module with regards to test and documentation.

c) MAPI: MAU Application Programmer's Interface. This is a library used by the application to access the services of the IEC 61162-4 protocol. The MAPI is part of the protocol module.

d) LNA: Local Network Administration. This module implements the IEC 61162-400 protocol and is part of the protocol module.

e) Local MAU n: The LNA can service several MAUs. The test procedure will require the testing of this ability, both as seen from the protocol and from the application module. A Local MAU is a test-application, linked together with a MAPI for the purpose of doing specific tests.

f) Remote LNA: The test-procedures will require the use of one or more LNAs located on one or more remote host computers. These LNAs can be of the same type as the LNA under test or some other (already tested) variant.

g) Remote MAU: The test-procedures will require the use of one or more test applications, connected to remote LNAs for the purpose of doing specific tests on protocol or application module.

6.2 System configurations

Each protocol function shall be tested in an appropriate number of system configurations. The number of configurations will depend on the function in question. The following clauses define the configurations used in this standard.

Note that the complex configurations include simpler configurations, for example a full test in a complex configuration will normally satisfy all requirements for simpler configurations. However, the order of testing indicated below will help in isolating potential errors in a given protocol implementation and as a principle, the most complex tests will only be performed in the low and medium complex configurations.

6.2.1 1L1M: One LNA with one MAU

The simplest configuration is where one LNA is tested with one local MAU. This test is used to test some aspects of MAU and interface management.

6.2.2 2L2M: Two LNAs and two MAUs

The most basic working configuration is a test where the local LNA is tested with a local MAU and a remote MAU on a remote LNA. The test shall be executed in two steps:

a) The local MAU acts as server and provides services to the remote MAU.

b) The local MAU acts as client and requests services from the remote MAU.

This will in general require that the MAUs are exchanged between the tests or that one MAU can be configured to act as both client and server.

In a configuration where both LNAs are identical, the first test will suffice for both steps, as the LNA (in the two instances) are used both in server and client configuration.

6.2.3 1L2M: One LNA and two MAUs

This is the same test as in the previous clause, but the LNA handles both MAUs. This tests the LNA's ability to handle locally established MAU-MAU connections and transactions.

6.2.4 3L6M: Three LNAs and six MAUs

In this configuration, the local and remote LNAs are equipped with two MAUs each. All but one of the local MAUs are clients and use the same service on the one local MAU that is configured as server. This topology tests that several requests from different clients are handled individually and not mixed during processing.

6.2.5 nLnM: Many LNAs with many MAUs

This is a large topology with several LNAs and at least one MAU on each. This topology shall test general robustness in a large system.

6.3 Test MAUs

This clause defines the basic functionality of the different test MAUs that shall be used to perform the tests.

6.3.1 Data representation

This MAU shall test correct handling of all types of data transformation between native and network format. It shall also test the correct handling of arrays.

The test shall be done by letting a test client transfer data to a test server and by letting the server check that the data is correct as expected. Note that it is not sufficient to send data to the server and back for a check there, as this may not detect errors that are done both in translation to and from the network format. It is recommended that the test MAUs use a script file to specify what data to transfer in what sequence and let both MAUs use the same configuration file to check correct operation.

The test MAU shall transfer all data types in arrays of odd length of at least three. This is to check that data alignment is handled correctly.

The test MAU shall check that variable length arrays with actual lengths less than, equal to and greater than the specified maximum lengths are handled correctly. The last case shall cause a run-time error at the sending side. All data types shall be tested as arrays.

The test MAU shall test the correct operation of the union type by creating a union with at least one element of each data type and sending repeated data messages, each with a different data type.

A possible specification for this MAU is listed in Clause A.2.

6.3.2 Function call tests

This MAU shall be used to test that all services supplied by the basic protocol are correctly implemented. It shall also be used to test multiple transactions and delayed transactions. Furthermore, it shall be used to test connections to full or partial interfaces.

The MAU shall have three interfaces with a selection of connection points, in total implementing all functions provided by the protocol.

Both the server and the client shall be able to check that a certain sequence of requests and acknowledgements is executed. It is suggested that each client in the test scenario is assigned a unique identity code that it uses, together with a request sequence code as its argument in all requests (indata). The server can respond with a per-session transaction sequence number (response) together with the client's input code (outdata) that can be used by the client to verify that the sequencing is correct.

The client, when doing the sequence verification must consider that different priority requests can be returned with different sequence numbers.

A possible specification for this MAU is listed in Clause A.3.

7 Test of general protocol modules

This clause contains the minimum test requirements for the basic protocol modules. These modules are the LNA, the MAPI and the T-profile. The tests will use the test MAUs and topology defined in clause 6.

Unless otherwise stated, all required responses shall be manifest after a very short time (significantly less than a second). A short delay means typically up to 5 s. The actual time will depend on the state of the involved LNAs and MAUs.

7.1 MAU session management

7.1.1 Connect a MAU to an LNA and test basic LNA functions

This clause defines a test that verifies that a MAU can connect to its LNA, that it gets registered there and the LNA starts the relevant watchdog service. The test uses the 1L1M configuration. The test may use any MAU, for example the data transfer test MAU described in 6.3.1. For the purposes of this test, the MAU need not export its interface.

The test shall be performed as follows:

a) Start LNA and then connect MAU. The MAU shall successfully connect and receive the connection grant message from the LNA. This test shall be run with three MAU configurations:

   1) Where a watchdog timeout is defined and a response function is implemented: It shall verify that the watchdog message from the LNA arrives as specified and that an answer to the message keeps the MAU alive.

   2) Where a watchdog timeout is defined and a response function is not implemented: It shall verify that the watchdog message from the LNA arrives as specified and that a missing answer to the message closes the MAU-LNA connection after the defined interval.

   3) Where no watchdog timeout is defined: Verify that no watchdog message arrives from the LNA.

b) Start the MAU when there is no LNA present. The MAU shall retry the connection attempt and/or print an error message.

c) Use scenario b), but after two MAU connect retries, start LNA and verify that the MAU connects to it as in step a).

7.1.2 Multiple MAU name handling

This test verifies that the LNA handles multiple MAUs correctly. Any two MAUs that can respectively act as server and client of a simple interface can be used for these tests, for example the data transfer test MAU described in 6.3.1. A 2L2M configuration shall be used for this test, but only one client/server configuration need be tested (e.g. a) below).

The test shall be run in the following sequence:

a) Connect a MAU named "A" to LNA 1. This MAU shall request a connection to MAU "B" that is not defined in the system.

b) Connect a new MAU named "A" to LNA 2. The new MAU shall be allowed to connect, but should get a duplicate name warning (possibly after a short delay).

c) Connect a new MAU named "A" to LNA 1. This MAU shall be denied access.

d) Connect MAU "B" to LNA 2. MAU "A" shall get a connection established.

e) Connect a MAU "B" to LNA 1. This shall receive a warning as a duplicate MAU.

f) Remove MAU "B" from LNA 2. LNA 1 shall immediately switch the connection of MAU "A" to the local MAU "B".

7.2 Interface management

7.2.1 MAU accept interface export

This test verifies that a MAU can export a server side interface description to the LNA. The test is run in the 1L1M configuration with any MAU that can export three identical interfaces with different interface names. The interfaces shall contain at least 10 connection points.

The test shall be executed in the following steps:

a) Export one interface description. This shall be accepted by the LNA.

b) Export one more instance of the same interface as in point a) (with a different interface name, but same connection point names and configuration). This shall be accepted by the LNA.

c) Export one more time the same interface as in item a), with all attributes identical. This shall be rejected by the LNA without causing any error.

d) Export one more instance of the same interface as in point a) (with a different interface name than in a) and b), but same connection point names and configuration). This shall be accepted by the LNA.

e) Remove the interface that was exported in item b). This shall be accepted by the LNA.

f) Kill and restart the LNA and run all points up to and including e): The MAU shall detect link failure and give notice of session loss, then it shall reconnect and respond as specified above for the different items. Run this sequence at least 10 times.

g) Kill and restart the MAU and run all points up to and including e): The LNA shall detect loss of MAU, clean up state and allow a new connection attempt. The responses shall be the same as in the first run. Run this sequence at least 10 times.

7.2.2 MAU connect interface management

This test shall verify that a client MAU can connect to a server. The configuration is 2L2M. The server and client MAU can be any MAU that can respectively export and import an interface with at least 10 connection points. The MAU must also be able to remove and add the interface at the tester's prompt.

The test shall be run in the following sequence:

a) Let the server MAU connect to LNA 1.

b) Let the client MAU connect to LNA 1. The connection shall be established, the server notified of the new client and it shall be possible to execute transactions.

c) Remove the server MAU: The client shall be notified of the loss and it shall not be possible to initiate transactions on the client side.

d) Reconnect the server MAU: The client shall be reconnected, the server notified of the new client and it shall be possible to execute transactions.

e) Remove the client MAU: The server shall be notified of the lost session.

f) Reconnect the client at LNA 2. The connection shall be established, the server notified and it shall be possible to execute transactions.

g) Remove the server MAU: The client shall be notified of the loss and it shall not be possible to initiate transactions on the client side.

h) Reconnect the server at LNA 2: The connection shall be established, the server notified and it shall be possible to execute transactions.

i) Let the server remove the interface: The client shall be notified and the connection broken.

j) Let the server add the interface: The connection shall be established again.

k) Let the client remove the interface: The connection shall be broken.

l) Let the client add the interface: The connection shall be established again.

7.2.3 MAU part interface connection management

This test shall verify that sub- and super-set interfaces are handled correctly. The test can use configuration 1L2M with a server MAU with at least 10 connection points and a client (or several clients) that has interfaces that are a sub-set, super-set and a variant of the interface provided by the server.

a) Connect the server MAU to the LNA.

b) Connect a client MAU with a sub-set interface to the LNA. The connection shall be established and it shall be possible to execute transactions.

c) Attempt to connect a client MAU with a variant interface to the LNA (one attribute shall be different, for example one MCP name or one format string). Verify that the interface cannot be connected to the server.

d) Connect a client MAU with a super-set interface to the LNA (one additional MCP has been added to the interface). Verify that the interface cannot be connected to the server.

7.2.4 MAU self-connect

This test shall verify that a MAU can connect to itself. The test shall use one LNA and one MAU. The MAU shall have at least one server interface and one client interface that can connect to each other. The function test MAU described in 6.3.2 can be used. The test shall be executed as follows:

a) The MAU shall export its server interface.
b) The MAU shall connect to itself through the client interface. The connection shall be established.
c) The MAU shall be removed.
d) The MAU requests a connection to its connect interface.
e) The MAU exports the server interface and the connection shall be established.
f) The MAU removes its server interface and then re-exports it. The connection shall be re-established.

All connection establishments shall be verified by sending a transaction on them (see also 7.3.2).

7.3 Transaction management

7.3.1 General transaction handling

Transaction handling is checked with a function call test MAU as described in 6.3.2 or similar. The same tests shall be performed in both the 1L2M and 2L2M configurations, except for the subscription tests which shall be performed in a "2L3M" configuration (two servers and one client, and one server residing on the same LNA as the client).

For each of the protocol's transaction types, the following requirements shall be verified:

a) Transaction: A reliable request shall be sent and it shall be verified that the corresponding acknowledgement is received or not (where no acknowledgement is expected).
b) Priority: It shall be verified that messages that are sent (requests or acknowledgements), with a specified priority, are received with the same priority.
c) Priority sequence: It shall be verified that two requests made immediately after each other with first a lower and then a higher priority, are acknowledged with the highest priority first. The result of this test will, however, depend on the load situation and the actual timing in the test scenario.
d) Delayed acknowledgement: The server shall delay the acknowledgement for two requests until two pending requests are registered with the client. The requests shall be acknowledged in the reverse order and the client shall be able to identify the reversal.
e) Subscription: Individual subscription acknowledges shall be sent one by one to each client. Other subscription acknowledges shall be copied by the LNA to all subscribers. Broadcast subscriptions shall be sent as one by the server LNA but copied by the receiving LNA. The test shall use two clients per server, one client on the same LNA and one on a remote LNA. It shall be verified that first and later acknowledgements are received as prescribed.
f) Cancel subscription: In the same configuration as normal subscription and after some normal acknowledgements have been received, one client shall cancel the subscription. It shall be verified that the cancelling client does not receive any more acknowledgements while the other client still does. It shall also be verified that the server gets the cancel message.

g) Cancel transaction: an acknowledgement shall be delayed by the server and the client shall cancel it. Thereafter, the client shall issue a new request. It shall be verified that the cancelled acknowledgement does not get delivered to the client and that the following acknowledgement is delivered.

h) State cleanup: One of the servers shall close its connected interface and then reconnect without starting a new subscription. It shall be verified that no acknowledgements are received by this server. The close of connection shall be tested by 1) Normal programmed close interface; 2) Closing the MAU; 3) Closing the server's LNA without closing the MAU and then restarting the LNA. Results shall in all cases be that the connection is established, but that transactions do not arrive.

The transaction types (function, read, write, non-acknowledged write, subscription, broadcast subscription, individual subscription and anonymous broadcast) shall be tested against the above requirements as listed in Table 1 below.

Table 1 – Transaction test requirements summary

	f	r	w	n	s	b	i	a
Transaction	x	x	x	x[a]	x[b]	x[b]	x	
Priority	x	x	x	x	x	x	x	x
Priority sequence	x	x	x	x				
Delayed acknowledgement	x	x	x					
Subscription					x[c]	x[d]	x[e]	x[f]
Cancel subscription					x	x	x	
Cancel transaction	x	x	x		x[g]	x[g]	x[g]	
State cleanup	x	x	x		x	x	x	

[a] No acknowledgements shall be delivered to client.

[b] First acknowledgement shall be sent reliably to subscribing client alone.

[c] All messages to all clients shall be identical and reliable.

[d] As [c], but some messages may be lost.

[e] All acknowledgements shall be directed to one specific client. The server must keep track of potentially clients.

[f] No session between client and server. No knowledge of each other shall be required.

[g] The cancellation shall stop subscription messages and the new request shall start it again.

7.3.2 Self-subscription

This test shall verify that a MAU can send transactions to itself. The test shall use one LNA and one MAU. The MAU shall have at least one server interface and one client interface that can connect to each other. The interface shall at least have one ordinary subscription MCP and one individual subscription MCP. The function test MAU described in 6.3.2 can be used. The test shall be executed as follows:

a) The MAU shall export its server interface.

b) The MAU shall connect to itself through its connect interface.

c) The MAU shall subscribe to both MCPs and the tester shall verify that acknowledgements are received correctly.

7.3.3 Multiple server/client subscription and session verification

This test shall verify that subscribe transactions are routed correctly in a complex configuration. The test shall use two LNAs and ten client MAUs, where each client MAU can subscribe to two server MAUs and where the server MAUs also supplies at least two sets of connectable interfaces. At least one interface shall be used in each connection and the interface shall have at least one individual and one normal subscription MCP. All interfaces (sets) shall be identical. The function test MAU described in 6.3.2 can be used. The topology shall be as illustrated below (only one half is illustrated, the connection sets shall be the same seen from both LNAs).

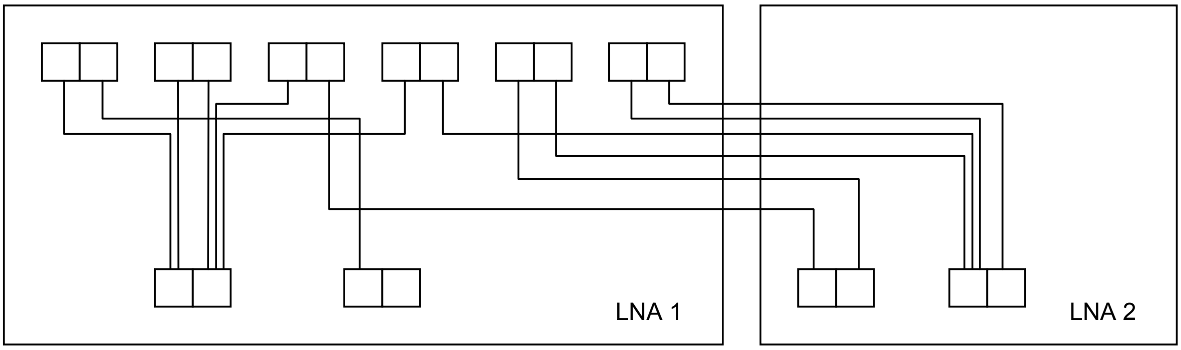

Figure 4 – Multiple client/server connections

The figure shows the client MAUs on top and the servers at the bottom. The connections are represented as lines. The LNAs are represented as boundaries around the MAUs they support. The following tests shall be run:

a) All MAUs shall connect to each other as specified.

b) The clients shall request subscriptions on the servers' MCPs (at least one normal and one individual per the two interfaces). The servers shall send acknowledgements regularly.

c) It shall be verified that the acknowledgements sent are received by clients as prescribed (ordinary subscribe duplicated to all subscribers, individual subscribe only to the acknowledged clients).

d) The clients shall verify that the session code following each transaction is the same as when the connection was established.

e) Each client shall cancel subscriptions on one of their MCPs and check that no more messages are received. They shall then resume subscription and verify that new acknowledgements arrive. The test shall be repeated for all MCPs.

These tests can be repeated with other types of MCPs, but it is assumed that the ordinary and individual subscription mechanisms are the most complex for the LNA and that other mechanisms are sub-sets of these.

7.3.4 Data marshalling

It shall be tested that a new implementation of the protocol translates data between native format and network format correctly. This requires the use of a MAU and an LNA with verified data translation capabilities. This can be established by inspection of received and sent network messages. It is also suggested that LNA and MAUs running on different computer architectures are used in these tests. The tests shall be run in a 2L2M configuration with a MAU capable of transmitting all data types as described in 6.3.1. The test shall be run as follows:

The server and the client MAUs reside on different LNAs. They shall use a transmission scheme agreed between them in a way other than through the binary protocol, for example by using a common text format configuration file or by transmitting a printed representation of the values as a text string. The receiver shall verify that the received value corresponds to the value that should be sent. Both sides shall transmit and receive all values. The following issues shall be tested:

a) All extreme values for each data type shall be transmitted. This includes zero, smallest non-zero number and largest number, in all applicable cases, including both negative and positive values. The values are listed in Annex B.

b) A sequence of values that are represented as an octet string with all octets having different values shall be transmitted to verify that correct octet sequencing is done. All sequences of this type shall be at least eight octets long, for example two w32 or four w16 values must be transmitted for these representation formats.

c) Variable length arrays and normal arrays with lengths of at least 8 shall be tested for correct order. All elements that are smaller than the computers word size shall be tested and a selection of other values shall also be tested. All length encoding formats for variable length arrays shall be tested.

d) Variable length arrays shall be tested for correct number of elements and order. It shall also be tested that it is not possible to transmit arrays that have a specified length which are longer than the one encoded in the format string. This shall cause an error return on the transmission side.

e) Records shall be tested for correct alignment between all reasonable combinations of elements (e.g. on a 32 bit computer, one should test w8, w16, w32 and w64 in different orders as well as arrays of w8 elements of less that four octets length).

f) Unions shall be tested for correct representation and correct enumeration of actual type transmitted. All different basic elements shall be tested as well as a selection of composite types, i.e. arrays and records. It is acceptable to create one union for this purpose.

7.3.5 Session identification

Session management tests shall prove that session authentication is possible over connections and transactions. The test shall verify operation in a 1L3M, 3L2M, 2L2M and 1L2M configuration. The MAUs used for the test shall have at least two interfaces where the MAU can check session codes for both connections and transactions. The client MAUs shall be able to connect to more than one server. The tests that shall be performed are:

a) The client MAU connecting on the two different interfaces shall get the same session code for both connection requests. This shall be tested for both client and server and for both one and two clients to the same server (see also d).

b) Transactions being sent by one MAU on several connection points on several interfaces shall be tagged with the same session code as on the interfaces during connect. This shall apply to client (for acknowledgements) and server (for requests).

c) A MAU that is disconnected and then connected again shall receive a different session code than that used for the first connection. This code shall also be verified for transactions (as in item b)).

d) Different client MAUs connected to one server shall be distinguishable by different session codes for interface connections and associated transactions.

e) Different server MAUs connected to by one client MAU shall be distinguishable by different session codes for connections and transactions.

7.3.6 Load limitation

The following checks shall be made to test load limitation facilities:-

a) A server MAU with one interface shall be defined with a maximum number of clients of respectively zero (no limit), one and greater than one. Tests shall be made to see that these limits are enforced by connecting one, two or more clients to the server.

b) A server MAU with an interface that has a client limit of one, or higher, shall be tested to see that the first higher numbered client is not denied a connection if the next highest numbered client has been denied.

c) An interface shall be defined with a maximum number of pending transactions of respectively zero (no limit), one and greater than one. Tests shall be made to check that these limits are enforced. This can be done by delaying the acknowledgement in the server.

7.3.7 Flow control

The following test shall be performed:

a) All types of flow control messages shall be sent on a local session and it shall be verified that the messages are received at the far side.

7.4 Exception handling

This clause specifies tests that are not directly related to functionality, but aimed at testing the response to various abnormal situations.

7.4.1 Message size limitations

A T-profile with or without a specified message size shall be tested as follows:

a) Send transactions and connection messages up to the message limit, or up to at least 50000 octets and verify that they are passed through the system.

b) Send a transaction message larger than the message limit (if defined) and verify that it is denied.

c) Do a connection attempt that require larger messages than maximum message size (if defined) and verify that the connection request is fragmented into more messages.

7.4.2 Overload handling

A MAU that stops receiving messages of low priority (both by sending a session stop message to the LNA or by stopping the low priority part of a T-profile) shall still be able to receive higher priority messages.

a) Verify that when the LNA gets a congested output link, it sends the appropriate session control message to connected MAUs.

b) Verify that low priority messages sent by remote MAUs are negatively acknowledged and returned to sender if they cannot be queued.

c) Verify that the congested MAU still receives urgent messages.

7.4.3 Redundancy

Tests shall be made so that it can be shown that no single fault causes loss of connection between two LNAs. These tests shall as a minimum include:

a) Loss of one cable.

b) Loss of one network interface card/connector.

c) Loss of one hub and/or router.

It shall be verified that a single fault has no functional consequences for the system (for urgent as well as normal priority messages) and that an appropriate error message is sent to the application level.

7.4.4 Loss of connection

Tests shall be made to verify that loss of connection is handled correctly. A lost connection between two LNAs shall cause messages to be sent to MAUs that are connected together via these two LNAs. The following cases shall be verified:

a) A remote computer is totally removed from the network (either by removing connectors or removing power).

b) A local computer is suddenly isolated from the network (by powering down the hub or removing the cable).

c) A remote LNA is suddenly killed and left dead for at least 10 s. It shall be verified that the link loss is detected.

d) A remote computer is run up to 100 % CPU load (by a dummy high priority program).

7.5 General high load tests

This subclause prescribes tests that shall be made to test and document high load performance.

7.5.1 Many modules

The system shall be tested by using a number of LNAs and MAUs that reflect worst case conditions for operation. A minimum scenario is 10 LNAs with 3 MAUs each. Each MAU shall both be configured as client and server and each server shall be connected to by at least two client MAUs.

The following tests shall be made (see details in previous clauses):

a) The system shall withstand the sudden loss of one or more MAUs. This shall be independent of the MUA being a server, client or combined server/client. Tests shall be made for at least three different MAUs where the most heavily loaded servers shall be included. When the lost MAUs are reconnected, the system shall resume normal operation.

b) The system shall withstand the loss of any number of LNAs. When LNAs are restarted, normal operation shall be resumed.

c) The system shall withstand loss of one or more host computers. When the host computers are restarted, normal operation shall be resumed.

Loss of components of the system shall be handled without loss of function and appropriate error messages shall be given to the operator.

The system shall in all cases resume normal operation when lost components are reconnected. This shall be verified by periodic transmission of transactions.

7.5.2 High load

The system shall be tested by sending as much data (number of octets per second) and number of messages (transactions per second) as [reflects] worst case conditions for operation in a normal configuration. As a minimum this shall be 10 transactions per second with 100 octets per message and two transactions per second with 3000 octets per transaction per MAU involved in the test. The following scenarios shall be tested:

a) 10 LNAs with three MAUs each sending data to each other in a system with a well-distributed load.

b) Two LNAs with 10 MAUs each sending data between each other (bi-directional transmission).

Round trip time for transactions shall be measured for normal and urgent transactions in both cases for normal network load (no other stations sensing data) and for high network load (some other station is loading the network to a defined level – minimum 25 % network utilisation for Ethernet type T-profile).

8 T-profile tests

The T-profile cannot usually be tested using only black-box techniques. However, the following clauses prescribe tests that shall be made, although the way tests are made may vary between implementations.

8.1 Peer-to-peer message networks

Peer-to-peer networks are used between LNAs. These shall support connection from any side and at least two priority levels.

Note that some T-profiles may offer high and urgent priority links as different instances (connection points). This must be considered during tests.

8.1.1 Normal connection management

It shall be checked that it is possible to establish a connection and to send both priority classes of messages.

It shall be verified that simultaneous connections on both redundant channels are handled correctly.

It shall be verified that any order of connection establishment on the two redundant channels is handled correctly.

8.1.2 Re-establishment tests

It shall be verified that loss of connection on any side does not inhibit re-establishment of the connection once the lost partner has been reconnected. This shall be tested at least 10 times in succession and it shall be verified that this does not lead to memory leaks.

8.1.3 Simultaneous connection establishment

It shall be tested that two computers connected to each other at the same time resolve their two connection attempts into one connection point at each side. This may be tested by inserting delays into the T-profile modules to check the operation of the state machine.

8.1.4 Loss of one priority channel

It shall be verified that loss of one priority channel is handled [gracefully] and that re-establishment of the channel is done correctly.

8.1.5 Loss of one redundant channel

It shall be verified that loss of one redundant channel is handled [gracefully] and that re-establishment of the channel is done correctly.

8.1.6 Measurement of channel capacity

A message shall be sent from one party and returned to the other party and the round trip time shall be measured. Both normal and urgent messages shall be used. The following cases shall be used:

a) Normal operation under low network and CPU load.

b) Normal operation under high network load (minimum 25 % network load for Ethernet).

c) Normal and urgent messages under high normal priority channel utilisation (minimum 50000 octets per second normal traffic for Ethernet).

8.2 Client-server message networks

Client-server message networks are used between MAUs and LNAs. These shall support [one or two] priority levels. These networks are not redundant.

8.2.1 Normal connection management

It shall be checked that it is possible to establish a connection and to send all priority classes of messages.

It shall be verified that any order of connection establishment on the two priority channels is handled correctly.

8.2.2 Re-establishment tests

It shall be verified that loss of connection on any side does not inhibit re-establishment of the connection once the lost partner has been reconnected. This shall be tested at least 10 times in succession and it shall be verified that this does not lead to memory leaks.

8.2.3 Loss of one priority channel

It shall be verified that loss of one priority channel is handled [gracefully] and that re-establishment of the channel is done correctly.

8.2.4 Measurement of channel capacity

A message shall be sent from one party and returned to the other party and the round trip time shall be measured. Both normal and urgent messages shall be used. The following cases shall be used:

a) Normal operation under low network and CPU load.

b) Normal operation under high network load (minimum 25 % network load for Ethernet).

c) Normal and urgent messages under high normal priority channel utilisation (minimum 50000 octets per second normal traffic for Ethernet).

8.3 Client-server stream networks

Client-server message networks are used between MAUs. These shall support low and normal priority levels. Normal connection management

8.3.1 It shall be checked that it is possible to establish a connection and to send the relevant priority class data.

It shall be verified that any order of connection establishment on the two redundant channels is handled correctly.

8.3.2 Re-establishment tests

It shall be verified that loss of connection on any side does not inhibit re-establishment of the connection once the lost partner has been reconnected. This shall be tested at least 10 times in succession and it shall be verified that this does not lead to memory leaks.

8.3.3 Loss of one redundancy channel

It shall be verified that loss of one redundant channel is handled without loss of function and that re-establishment of the channel is done correctly.

8.3.4 Measurement of channel capacity

A stream of at least 100000 octets shall be sent from one party and when fully received, shall be returned to the first party and the round trip time shall be measured. Both normal and urgent streams shall be used. The following cases shall be used:

a) Normal operation under low network and CPU load.

b) Normal operation under high network load (minimum 25 % network load for Ethernet).

c) Normal and urgent streams under high normal priority channel utilisation (minimum 50000 octets per second normal traffic for Ethernet). This requires the establishment of at least two channels.

8.4 Broadcast networks

Broadcast networks are used between LNAs. These shall support one priority level. These networks are redundant.

8.4.1 Normal operation tests

It shall be verified that messages can be sent and received by all parties. It shall be verified that messages sent by one party also are received by this party.

8.4.2 Loss of one redundancy channel

It shall be verified that loss of one redundant channel is handled [without loss of function and that re-establishment of the channel is done correctly.

8.4.3 Measurement of channel capacity

Messages shall be sent from one party and when received, shall be returned to the first party and the round trip time shall be measured. Both normal and urgent messages shall be measured, if supported by the network. The following cases shall be used:

a) Normal operation under low network and CPU load.

b) Normal operation under high network load (minimum 25 % network load for Ethernet).

9 Test requirements for applications

Applications (MAUs) that use the PISCES protocol to communicate with other applications require a companion standard specification to define their interfaces to other MAUs. The tests listed in this clause shall verify that the companion standard is defining the actual interface.

In addition to the tests listed in this clause, the basic protocol software shall have passed all relevant tests as listed in previous clauses.

9.1 Companion standard specification

The application's companion standard specification shall be tested for compliance to the rules specified in the companion standard general guidelines (IEC 61162-420). This is usually most conveniently done with the help of a computer tool.

Note that correct behaviour of applications is specified in the function block specification. Many of the following clauses will therefore refer to this in the test requirements.

9.1.1 Syntactic correctness

The documents shall be checked for syntactic correctness so that a computer tool can use them as the basis for code generation.

9.1.2 Completeness

The specification shall be checked for completeness as follows:

a) All interfaces and all connection points shall be specified and named.

b) [Name of MAU] shall be specified.

c) Load limitation parameters shall be specified where applicable.

d) Password and authentication requirements shall be specified where appropriate.

e) Any configuration possibilities shall be specified.

NOTE Configuration possibilities will normally include at least MAU name and commonly also interface names. If passwords are in use, these should normally also be configurable.

f) A function block description for the application shall be included. This description shall describe the behaviour of the MAU as a result of all normal stimuli, including loss of communication and other common exceptions.

9.1.3 Foundation class relationship

It shall be checked that the specification places the application in the foundation class hierarchy where appropriate and that the specification adheres to the requirements (e.g. interface requirements) that comes from such a placement.

9.2 Interface correctness

The interfaces of the application shall be checked against the companion standard specification for full adherence to that. This will normally be done with a computer based testing tool.

9.2.1 Connection management

It shall be tested that the application can connect to the number of clients and servers that is specified:

a) It shall be tested that all servers are connected when they become available.

b) It shall be tested that the application reconnects (where applicable) to servers that are lost.

c) It shall be tested that the application allows connection from any number of clients that it is specified to support. It shall also be tested that any connection limitation functions, including password, work as specified.

d) It shall be tested that the application tolerates and handles loss of clients in a safe manner.

NOTE As a minimum, one should normally require that a client that is lost and then reconnects at least is treated as if it connected for the first time (e.g. complete reset of internal state). Other behaviours may be specified.

e) It shall be tested that the application tolerates loss of own LNA in a safe manner and that it reconnects to all clients and servers automatically (if the application does not specify other behaviour).

9.2.2 Session handling

It shall be tested that the application handles session control and information correctly. As a minimum, the following shall be tested:

a) The application shall, if specified, check the authenticity of client and server data by using an authentication mechanism. It shall reject unauthorised requests.

b) The application shall respond to session control messages (flow control) from LNA or connected MAUs.

c) The application shall handle rejected requests in a safe manner.

9.2.3 Transaction management and functionality tests

The application shall be able to handle all relevant transaction types. As a minimum, the following shall be tested:

a) All connection points in all interfaces shall be tested to verify that data transmitted is correctly received and sent. The test need not be exhaustive, but a suitable selection of data values shall be used.

b) It shall be verified that the application gives appropriate response to stimuli. This shall be tested with a detailed test and expected response scenario, with a minimum configuration of clients and servers. This test shall cover all relevant connection points and all relevant sequences of stimuli as well as the most common exceptions, for example loss of remote MAU or own LNA.

c) It shall be tested that the application gives individual responses to different servers and/or clients when appropriate. This test need not be exhaustive over all connection points, but a representative selection shall be made. The test shall use more than two connected remote MAUs when applicable.

9.3 High load tests

Applications that specify load limiting functionality (transaction or connection limitations) shall be tested to see if these mechanisms function correctly.

Applications shall also be tested for operations under worst case operations (normally a high number of transactions, but also possibly long messages).

10 Documentation requirements for general protocol modules

10.1 General software and test documentation

The modules shall be documented according to the selected development method and according to standards that are used during the development process (see 4.3). These documents shall be maintained by the developer and be available to the authority that is responsible for verification that the developer is doing its work in adherence to the relevant development standard. If such an authority does not exist, then the developer must be able to make the documentation available to any users of the protocol modules.

10.2 Technical specifications

The technical specification for the modules shall at least contain:

a) Version codes and compilation dates for all components of the modules.

b) Version code for the protocol that is implemented.

c) Specification of physical platform supported, including hardware and software versions used.

d) Type of interfaces supported. This shall include all possible mechanisms available for connection to other LNAs and served MAUs (T-profiles).

e) Performance test results. It shall be documented how many MAUs can be supported and the maximum throughput, in terms of transactions and octets per second.

f) Limitation in services provided: If the modules have an upper limit on message length that is supported, this shall be documented. Other limitations that may be of interest to the user shall also be documented.

This documentation shall be made available to the entity responsible for the integration of the application into a system.

11 Documentation requirements for applications

11.1 General software and test documentation

The same requirements as in 10.1 also apply to applications.

11.2 Companion standard specification

An application shall be documented with a companion standard specification. This documentation shall be created and verified according to the requirements in 9.1.

These documents shall be made available to the entity responsible for the integration of the application into a system.

11.3 Technical specification

The technical specification for the application shall at least contain:

a) Version codes for the modules in the application.

b) Version code for the protocol that the application implements and is tested against.

c) Specification of the physical platform used or required, including hardware and software versions.

d) Class of application; i.e. if it requires an external LNA, if it can coexist with other applications on one computer, or if it contains an LNA that will accept service requests from other MAUs).

e) Special requirements for LNA if not included: Type of T-profile expected, special services from LNA (e.g. requires use of LNA-MAU).

f) Maximum load offered to other MAUs and the LNA in transactions per second and octets per second. It shall be specified if the application will accept load limitation control messages (session control messages).

g) Maximum load tolerated from other MAUs in transactions and octets per second. It shall be specified if this upper limit is enforced by load limiting functions (session control functions).

h) Limitation in services provided: If the modules have an upper limit on message length that is supported, this shall be documented. Other limitations that may be of interest to the user shall also be documented.

This documentation shall be made available to the user.

Annex A
(informative)

Companion standard specifications

A.1 Introduction

This annex contains companion standards for MAUs that can be used as test MAUs. These are only suggested MAUs and are included to illustrate principles in test requirements.

A.2 Data transfer test

```
INTERFACE DataTransfer
 * This document contains the specification of a test MAU for testing correct
  data translation.

 * Revision history:
   000108 1.0 First IEC CDV release

 VERSION   1.0
 DATE    2001-01-08
 RESPONSIBLE IEC TC80/WG6

USAGE
 The client is expected to send a sequence of messages that shall be checked
 by the server for correctness. A test script will be used to determine what
 data to send.

;-------------------------------------------------------------
DATA TYPES

  DATA BLOCK ArrayData
   [word8_m:13]bool_m    bData
   [word16_m:13]word8_m   w8Data
   [word32_m:13]word16_m   w16Data
   [word8_m:13]word32_m   w32Data
   [word16_m:13000]word64_m w64Data
   [word32_m:13]int8_m    i8Data
   [word8_m:13]int16_m    i16Data
   [word16_m:13]int32_m   i32Data
   [word32_m:13000]int64_m  i64Data
   [word8_m:13]char8_m    c8Data
   [word16_m:13000]float64_m f64Data
   [word32_m:13]float32_m  f32Data

  UNION UnionData: word16_m
   bool_m  bData
   word8_m  w8Data
   word16_m w16Data
   word32_m w32Data
   word64_m w64Data
   int8_m  i8Data
   int16_m  i16Data
   int32_m  i32Data
   int64_m  i64Data
   char8_m  c8Data
   float64_m f64Data
   float32_m f32Data

;-------------------------------------------------------------
CONNECTION POINTS
```

```
;------------------------------------------------------------
 WRITE SendArray
 * Send an array data type.

  INPUT
   ArrayData arrayData

  * Precondition
   none

  * Postcondition
   none.

  * Informal Explanation
   Server will check against own records if data is all right.

;------------------------------------------------------------
 WRITE SendUnion
 * Send a union data type

  INPUT
   UnionData  unionData

  * Precondition
   none

  * Postcondition
   none

  * Informal Explanation
   Server will check against own records if data is all right.
```

A.3 Function call test

```
INTERFACE FunctionsOne
 * This document contains the specification of a test MAU for testing correct
  function invocations.

 * Revision history:
   000108 1.0 First IEC CDV release

 VERSION   1.0
 DATE    2001-01-08
 RESPONSIBLE IEC TC80/WG6

USAGE
 The client is expected to send a sequence of transactions that are responded to
 by the server. The execution and checks of requests and acknowledgements are
 controlled by a script file.

;--------------------------------------------------------------
DATA TYPES
  none

;--------------------------------------------------------------
CONNECTION POINTS

;------------------------------------------------------------
 READ ReadData
 * Read data.

  OUTPUT
   word32_m  response
   word32_m  dataout

  * Precondition
   none
  * Postcondition
   none.

  * Informal Explanation
   none.

;------------------------------------------------------------
 WRITE WriteData
 * Write data.

  INPUT
   word32_m  datain

  * Precondition
   none

  * Postcondition
   none

  * Informal Explanation
   none.

;------------------------------------------------------------
 FUNCTION FunctionData
 * Write data.

  INPUT
   word32_m  datain

  OUTPUT
   word32_m  response
   word32_m  dataout

  * Precondition
   none

  * Postcondition
   none

  * Informal Explanation
   none.

;------------------------------------------------------------
 NONACKED WRITE WriteData
 * Write data.

  INPUT
   word32_m  datain

  * Precondition
   none

  * Postcondition
   none

  * Informal Explanation
   none.



INTERFACE FunctionsTwo
 * This document contains the specification of a test MAU for testing correct
   function invocations.

 * Revision history:
   000108 1.0 First IEC CDV release

 VERSION   1.0
 DATE    2001-01-08
 RESPONSIBLE IEC TC80/WG6
```

```
USAGE
 The client is expected to send a sequence of transactions that are responded to
 by the server. The execution and checks of requests and acknowledgements are
 controlled by a script file.

;--------------------------------------------------------------
DATA TYPES
  none

;--------------------------------------------------------------
CONNECTION POINTS

;------------------------------------------------------------
 SUBSCRIBE SubscribeData
 * Read data.

  OUTPUT
   word32_m  response
   word32_m  dataout

  * Precondition
   none

  * Postcondition
   none.

  * Informal Explanation
   none.



;------------------------------------------------------------
 INDIVIDUAL SUBSCRIBE ISubscribeData
 * Write data.

  INPUT
   word32_m  datain

  OUTPUT
   word32_m  response
   word32_m  dataout

  * Precondition
   none

  * Postcondition
   none

  * Informal Explanation
   none.

;------------------------------------------------------------
 BROADCAST SUBSCRIBE BSubscribeData
 * Read data.

  OUTPUT
   word32_m  response
   word32_m  dataout

  * Precondition
   none

  * Postcondition
   none.

  * Informal Explanation
   none.
```

```
INTERFACE FunctionsThree
 * This document contains the specification of a test MAU for testing correct
  function invocations.

 * Revision history:
   000108 1.0 First IEC CDV release

 VERSION   1.0
 DATE    2001-01-08
 RESPONSIBLE IEC TC80/WG6

USAGE
 The client is expected to send a sequence of transactions that are responded to
 by the server. The execution and checks of requests and acknowledgements are
 controlled by a script file.

;-------------------------------------------------------------
DATA TYPES
  none

;-------------------------------------------------------------
CONNECTION POINTS

;-----------------------------------------------------------
 ANONYMOUS BROADCAST SendData
  * Used to send a request.

  OUTPUT
   word32_m   indata

  * Precondition
   none

  * Postcondition
   none.

  * Informal Explanation
   none.


 ;-----------------------------------------------------------
 ANONYMOUS BROADCAST GetData
  * Used to acknowledge a request.

  OUTPUT
   word32_m  response
   word32_m  dataout

  * Precondition
   none

  * Postcondition
   none.

  * Informal Explanation
   none.
```

Annex B
(informative)

Examples of numeric values for tests

B.1 Extreme values

Extreme numeric values can be retrieved in ANSI C or C++ by using the system include file "limits.h". Table B.1 lists the constant values found in this file for a system that uses the same representation as the IEC 61162-4 standard. Note that actual values for floating point data may vary a bit between implementations, but that the ones presented here should be useful for the tests in this standard.

Table B.1 – Extreme values

Type	Limit	Value
bool_m	MIN	0
	MAX	1
char8_m	(see note 1)	
char16_m	(see note 1)	
word8_m	MAX	256
word16_m	MAX	65535
word32_m	MAX	(2147483647L * 2UL + 1)
word64_m	MAX	(9223372036854775807LL * 2ULL + 1)
int8_m	MIN	–128
	MAX	127
int16_m	MIN	–32768
	MAX	32767
int32_m	MIN	–2147483648L
	MAX	2147483647L
int64_m	MIN	–(9223372036854775807LL + 1)
	MAX	9223372036854775807LL
float32_m	MIN	–3.402823466385e + 38
	SMALLN	–1.175494350822e – 38
	SMALLP	1.175494350822e – 38
	MAX	3.402823466385e + 38
float64_m	MIN	–1.797693134862e + 308
	SMALLN	–2.225073858507e – 308
	SMALLP	2.225073858507e – 308
	MAX	1.797693134862e + 308

NOTE 1 The extreme values for character types depend on the host computer. They are either identical to the corresponding integer or word type, dependent on character signed or not.

NOTE 2 The integer values presented here are accurate. The smallest representable values are respectively 1 and minus 1.

NOTE 3 The floating point values are to be understood as the smallest and largest values representable on the network. They are not exact as some rounding errors occur when they are translated into decimal notation.

B.2 Non-interchangeable values

It is suggested that the different numeric values are built up by simply incrementing an octet value counter with one for each new octet. The sender can then print the proper numeric value out in its own representation and the printout is used by the receiver to verify that the numbers are the same. It is also possible to transfer the printed numeric value as a text string between sender and receiver.

Note that the number of significant decimals (not including sign) for the various numeric types is as described in Table B.2.

Table B.2 – Significant decimals

Type	Significant decimals
bool_m	1
char8_m	3
char16_m	5
word8_m	3
word16_m	5
word32_m	10
word64_m	20
int8_m	3
int16_m	5
int32_m	10
int64_m	20
float32_m	8
float64_m	15

ISBN 2-8318-XXXX-X

ICS 47.020.70

Typeset and printed by the IEC Central Office
GENEVA, SWITZERLAND